# Feedback — Week 1 Quiz

Help Center

You submitted this quiz on **Thu 7 May 2015 9:17 PM PDT**. You got a score of **40.00** out of **40.00**.

## Question 1

Which of the following describe what an Android Service is, according to the videos in week #1:

| Your Answer | | Score | Explanation |
|---|---|---|---|
| ☐ A Service is an application component that responds to system-wide broadcast announcements. | ✔ | 1.00 | |
| ☐ A Service is an application component manages access to a structured set of data by encapsulating the data and providing mechanisms for defining data security. | ✔ | 1.00 | |
| ☑ A Service is an application component that can perform long-running operations in the background and does not provide a user interface. | ✔ | 1.00 | |
| ☐ A Service is an application component that provides a single, focused thing to the user. | ✔ | 1.00 | |
| Total | | 4.00 / 4.00 | |

**Question Explanation**

Please see the video on "Overview of Android Started and Bound Services" from week #1.

## Question 2

Which of the following are key differences between a Started Service and a Bound Service,

according to the videos from week #1:

| Your Answer | | Score | Explanation |
|---|---|---|---|
| ☑ A Bound Service runs only as long as a client is bound to it, whereas a Started Service can run indefinitely. | ✔ | 1.00 | |
| ☐ A Bound Service is launched on demand, whereas a Started Service is pre-launched at during system boot sequence. | ✔ | 1.00 | |
| ☑ A Bound Service offers a client/service interface that allows extended two-way conversations between clients and the service. | ✔ | 1.00 | |
| ☐ A Bound Service always runs in the same process as its client, whereas a Started Service always runs in a separate process. | ✔ | 1.00 | |
| Total | | 4.00 / 4.00 | |

**Question Explanation**

Please see the video on "Overview of Android Started and Bound Services" from week #1.

# Question 3

Which of the following best describe the purpose of the onStartCommand() hook method, according to the videos from week #1:

| Your Answer | | Score | Explanation |
|---|---|---|---|
| ☐ It is used to notify a Service that it's being removed and should therefore cleanup any resources it holds. | ✔ | 1.00 | |
| ☑ It's often used in conjunction with the concurrency model the Service applies to perform its processing. | ✔ | 1.00 | |
| ☑ It is called each time a Started Service is sent an Intent from a client via a call to startService(). | ✔ | 1.00 | |
| ☐ It is dispatched by the Android Service framework when a Started Service is first launched. | ✔ | 1.00 | |

| | |
|---|---|
| Total | 4.00 / 4.00 |

**Question Explanation**

Please see the video on "Overview of Android Started and Bound Services" from week #1.

# Question 4

Which of the following are reasons why the stopSelf() method is passed the startId parameter, according to the videos from week #1:

| Your Answer | | Score | Explanation |
|---|---|---|---|
| ☐ To ensure that all the resources allocated by the onCreate() hook method are deallocated in the onDestroy() hook method. | ✔ | 1.00 | |
| ☐ To allow the Service to shut itself down and avoid running in the background and consuming system resources indefinitely. | ✔ | 1.00 | |
| ☐ To eliminate the need to explicitly acquire and release locks in critical sections. | ✔ | 1.00 | |
| ☑ To avoid prematurely terminating a Service while it is still processing concurrent Intent requests. | ✔ | 1.00 | |
| Total | | 4.00 / 4.00 | |

**Question Explanation**

Please see the video on "Programming Started Services" from week #1.

# Question 5

Which of the following are IPC mechanisms Android supports for communicating from a Service to an Actiivty, according to the videos in week #1:

| Your Answer | Score | Explanation |
|---|---|---|

| | | |
|---|---|---|
| ☑ Use an AIDL-based callback object passed from the Activity to the Service | ✔ | 1.00 |
| ☐ Send an Intent command via startService() | ✔ | 1.00 |
| ☑ Use a Messenger passed from the Activity to the Service | ✔ | 1.00 |
| ☐ Send an Intent command via bindService() | ✔ | 1.00 |
| Total | | 4.00 / 4.00 |

**Question Explanation**

Please see the video on "Activity and Service Communication" from week #1.

# Question 6

Which of the following are reasons why a Handler can't be used for direct communication between Activities and Services, according to the videos from week #1:

| Your Answer | | Score | Explanation |
|---|---|---|---|
| ☑ Handlers don't implement the Parcelable interface | ✔ | 0.80 | |
| ☑ Handlers can't be passed as "extras" to an Intent | ✔ | 0.80 | |
| ☐ Handler's can't be used to communicate between different threads | ✔ | 0.80 | |
| ☐ If a user has access to resource on the platform, all applications that the user launches have access to that resource | ✔ | 0.80 | |
| ☐ Handlers implement the Command Processor pattern, which only works within a single process | ✔ | 0.80 | |
| Total | | 4.00 / 4.00 | |

**Question Explanation**

Please see the video on "Activity and Service Communication with Messengers" from week #1.

# Question 7

Which POSA pattern(s) are associated with the use of Android Intents, Started Services, and Messengers, according to the videos in week #1:

| Your Answer | | Score | Explanation |
|---|---|---|---|
| ☑ Command Processor pattern | ✔ | 1.00 | |
| ☑ Active Object pattern | ✔ | 1.00 | |
| ☐ Half-Sync/Half-Async pattern | ✔ | 1.00 | |
| ☑ Activator pattern | ✔ | 1.00 | |
| Total | | 4.00 / 4.00 | |

**Question Explanation**

Please see the video on "Activity and Service Communication with Messengers" and "Programming Started Services" from week #1.

# Question 8

Which of the following are the meaning of onStartCommand() returning the START_NOT_STICKY value, according to the videos in week #1:

| Your Answer | | Score | Explanation |
|---|---|---|---|
| ☑ Service should remain stopped until explicitly started by some client code | ✔ | 1.00 | |
| ☐ Stop the Service and return an error code to the component that invoked it | ✔ | 1.00 | |
| ☐ Restart the Service via onStartCommand(), but don't redeliver the Intent (instead, pass null) | ✔ | 1.00 | |
| ☐ Restart the Service via onStartcommand, supplying the same Intent as was delivered this time | ✔ | 1.00 | |

| | |
|---|---|
| Total | 4.00 / 4.00 |

**Question Explanation**

Please see the video on "Programming Started Services" from week #1.

# Question 9

Which of the following patterns are applies in the Download application, according to the videos in week #1:

| Your Answer | | Score | Explanation |
|---|---|---|---|
| ☐ Chain of Responsibility | ✔ | 1.00 | |
| ☑ Factory Method | ✔ | 1.00 | |
| ☐ Bridge | ✔ | 1.00 | |
| ☑ Command Processor | ✔ | 1.00 | |
| Total | | 4.00 / 4.00 | |

**Question Explanation**

Please see the video on "Programming Started Services" from week #1.

# Question 10

Which of the following are ways that a Started Service can be shutdown, according to the videos in week #1:

| Your Answer | | Score | Explanation |
|---|---|---|---|
| ☐ The Service will be shutdown automatically when all clients unbind from it | ✔ | 1.00 | |
| ☐ The Service will shutdown automatically when there are no more Intents to process | ✔ | 1.00 | |

| | | |
|---|---|---|
| ☑ It can call stopSelf() to shut itself down | ✔ | 1.00 |
| ☑ Another component can shut down a Service by calling stopService() | ✔ | 1.00 |
| Total | | 4.00 / 4.00 |

**Question Explanation**

Please see the video on "Programming Started Services" from week #1.